From Images to Schemas

Stéfane Paris

2009 First International Conference on Intensive Applications and Services

From Images to Schemas

Stéphane Paris
Laboratory of Theoretical and Applied Computer Sciences
Paul Verlaine University, Ile du Saulcy, 57000, Metz, France
Stephane.Paris@ieee.org

Abstract

In Content Based Image Retrieval (CBIR), images are segmented to synthesize image information. Among several characteristics like color or edges, texture is useful for segmenting. This paper proposes an intensive multiresolution approach to texture segmentation based on a wavelet transform. The technique delivers schematic descriptions of images. That is to say, it provides the main regions of interest (ROIs) according to image information. Firstly, the process divides images into 2×2 blocks. Then, it tracks texture through the multiresolution offered by the wavelet transform to form featuring vectors. Next, a K-means algorithm partitions the texture vector space into clusters. Finally, a connected component extraction delivers the image schema.

keywords: CBIR, intensive schematization, texture, color, wavelet

1 Introduction

With the exponential growing of digital databases, multimedia search becomes more and more complex. Thus, organizing these data ensure the effectiveness and efficiency of their retrieval.

Concerning image retrieval (IR), a simple way to do is to annotate them. But these annotations are not always without ambiguity and are often manually done.

To improve IR, images can be identified by the information their pixels deliver. This is known as *content based image retrieval* (CBIR).

This paper deals with CBIR where images are described according to the objects they contain. Thus, pixels are labeled according to the objects they describe.

The presented work belongs to this category.

1.1 Related work

CBIR implies the need of models and tools to extract relevant information from image pixels to allow the matching of a requested image (or parts of it) with images in a database in a way to retrieve similar ones.

Different models define *texture* and *color*. Since texture is based on a repetitive local pattern, it is extracted from the luminance component of the images. So, images are first defined into a specific color space – e.g., Lab, YUV, YCbCr or Gaussian color space – where the first component delivers the luminance information. The two other components describe the chrominance.

The Lab color space need the specification of its white color and its conversion from RGB color space is non linear. Nevertheless, the euclidean distance can be used as a measure of color difference. Other more recent color spaces (Gaussian color space) have been proposed [8, 9]. Like Lab color space, assumptions are made that imply the specification of some parameters which determine the effectiveness of the color space.

By definition, texture is a *deep-structure* property [13]. So multiscale approaches are used to identify it [15, 21]. One of the most attractive is the wavelet analysis, since its main algorithm is effective and relatively easy to implement [2, 11].

Once the models of texture and color specified, the segmentation is done. There are two major approaches: *point of interest* and *regions*.

1.1.1 Point of interest

Points of interest are identified using tools like Harris point extractor[10]. Then, the selected points are labeled according to some measure concerning their neighborhood. For example, *Local Jet* and *SIFT* characterize interest points by vectors invariant under some geometric transformations [13, 7, 16, 17, 27].

However, the meaning of such set of points is not clear for CBIR user when she/he wants to retrieve a specific object or person in the image database (IDB).

978-0-7695-3585-2/09 $25.00 © 2009 IEEE
DOI 10.1109/INTENSIVE.2009.16

1.1.2 Regions

Here, an image is partitioned into *regions* that correspond to the objects present in the observed scene. At first glance, regions are more closed to the user meaning when queries are based on objects (or parts of them) present in the images. Unfortunately, their extraction is not trivial and their characterization suffers from invariance. In fact, regions are more sensible than points of interest to viewpoint, occlusions, scene clutter, and so on.

Three main classes of region extraction tool can be identified :

- The *split-and-merge* techniques decompose the image into homogeneous regions and then merge similar neighboring regions.
- The *graph-cut algorithm* maps a graph onto the image. The edges of the graph are weighted according to some pixel characteristic distances. Then the algorithm splits the graph into sub-graphs following the minimum edges [20, 25, 24].
- The *variational approaches* model the image by a piecewise continuous function. A functional is design according to some desired properties concerning the function (e.g., regularity) and the image regions (e.g., homogeneity in texture).
  Usually, the functional is minimized using some technique of optimization. An initial region partition of the image can be used to decrease the optimization technique cost [1, 14].

Despite that it is an expensive process and that the obtained regions do not provide invariance property, region extraction is attractive. It is because regions are actually significant to people when they search IDB.

The variational and cut-graph approaches are currently the most effective algorithm since they fit rather well objects in images in most configuration. Their main drawback is their algorithmic complexity.

Some hybrid approaches propose a mixture of interest points and regions. For example, interest points are first extracted. Then, local area (small regions) around the interest points are approximated as planar. By this way, the characterization can be defined invariant to some geometric transformation [17].

1.2 The proposed technique

The technique described in this paper proposes a region based image partitioning. The proposed labeling process is closed to the Blobworld system [4] and to the SimpliCITY system [23]. These systems describe pixels with some feature vectors to group them into regions. An image request is then based on user selection of regions to find in other images.

Blobworld extracts object regions using a steerable filter bank for texture. The pixels are characterized by vectors which are partitioned using the Expectation-Maximization (EM) algorithm with a mixture of K gaussians. The number of clusters, K, is first estimated by maximizing the likelihood. Then, a connected component process groups pixels into regions. However, the labeling process is expensive. EM and steerable filter bank are effective but difficult to use.

The common framework with the presented work is the separation between the pixel labeling and the connected component extraction. But, the present paper proposes a more intensive way to characterize pixels.

SimpliCITY uses wavelet transform (Haar or db4 wavelet) to characterize 2×2 blocs of pixels. These blocs are labeled using the standard K-mean algorithm where K is gradually incremented until the distortion between two successive values of K is no longer significant. The requested image is matched to the images of the database using the *Integrated Region Matching* (IRM) which is a many-regions-to-many-regions matching technique. Based on the χ^2 test, SimpliCITY also proposes the creation of an *ontology* where images are annotated as textured or not, as being a photography or a graphic. This ontology is used to prune the database before performing the IRM search.

As SimpliCITY does, the technique proposed in this paper uses the K-mean algorithm to classify pixel vectors which are the coefficients of a Haar wavelet analysis. The number K of clusters is predicted and then adjusted according to the relative scattering information. Since no spatial information is taken into account, a connected component process groups pixels into regions. While SimpliCITY stops at the first level of decomposition, the proposed wavelet analysis operates a decomposition up to 4. This is motivated by the fact that texture is a deep-structure [13, 2, 19, 11, 22].

The originality of this process is its simplicity with performances quite similar to those of more complex systems. Moreover, note that :

- The process could be used with compressed images too (e.g, JPEG2000 images) [12].
- This crude but intensive segmentation can later be improved using some level sets technique [1].

These two properties help achieve the CBIR goal : it is an intensive schematization of the given image which should be a good initialization for segmentation.

1.2.1 Principle

The principle is described in Figure 1. First, conversion from RGB color space to Lab color space separates luminance (L component) from chrominance (a and b components). As texture is a luminance characteristic, the following stages deal with the L component.

Texture is identified and characterized using a wavelet packet decomposition. Once this is done, each pixel is described by a characteristic vector. These texture vectors are partitioned into clusters with the help of a dissimilarity measure. Finally, a region is defined by a group of connected pixels which share the same texture labeling.

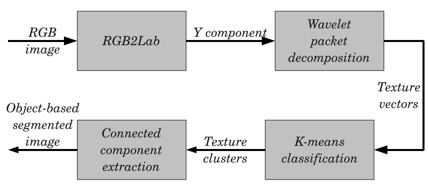

Figure 1. Schematizing process.

1.3 Plan

Section 2 presents the wavelet packet decomposition process which delivers the vectors characterizing the image texture. Then, sections 3 and 4 explain the two stages segmentation technique based on the texture vectors. Before concluding, section 5 analyzes several experiments.

2 Image characterization

Texture can be defined as a *repetitive pattern* [14, 5]. This pattern can be modeled in a multi-scale way describing it at different levels of observation. Roughly speaking, each level corresponds to a certain degree of precision in the observation. Thus, a fine-to-coarse characteristic stack is assigned to the pattern.

The *wavelet packet transform* decomposes the original image into four sub-images. One of them is an approximation of the original image. The three others provide the horizontal, vertical and diagonal details lost by the approximation. The process is reversible : combining together the four sub-images delivers the original image.

The process uses a bank of two 1D filters to separate low frequencies from high frequencies. The bank is applied successively on the lines and the columns of the image. A sub-sampling by two is done to eliminate the redundant information. Experiments use the Haar filters : $L(i) = \frac{x_i + x_{i+1}}{\sqrt{2}}$ and $H(i) = \frac{x_i - x_{i+1}}{\sqrt{2}}$
The reasons are the short length of the filters (the size of the filters) and their simplicity. The length of the L and H sequences is half the length of the sequence x.

The process is applied on each sub-images until a given threshold, J, of decomposition is reached. At the end, the decomposition can be presented as a tree where the root is the original image and the leaves the sub-images of the deepest decompositions (cf. fig. 2).

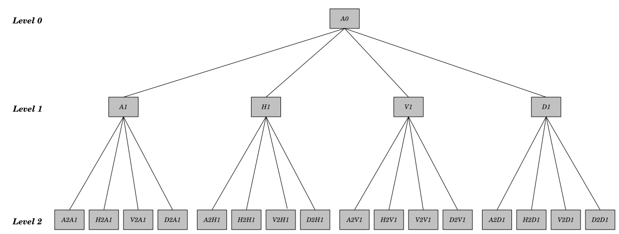

Figure 2. Wavelet tree with $J = 2$: $A0$ is the original image, $A1$ its approximation at level 1, $H2V1$ the horizontal detail at level 2 of $V1$ (the vertical detail at level 1), and so on.

To lighten this description, the tree can be pruned according to some energy measure. If the energy entropy of an internal node (a sub-image which is decomposed) is less then the sum of the energy entropies of its chlild nodes, the internal node becomes a leaf and the decomposition is pruned. Otherwise, the decomposition is kept.

Once the decomposition tree is pruned, the leaf image coefficients are grouped to form the characteristic vectors. The multi-scale description, delivered by the decomposition, implies that several pixels of the original image share some coefficients : for example, a coefficient of the approximation at level 1 ($A1$) is shared by four pixels of the original image. This encloses the third dimension of the texture patterns.

Experiments show that parameter J can be set to 3. When $J = 4$, the deepest decomposition level coefficients are shared by $4^J = 256$ pixels. This is, obviously, larger than texture currently observable into images.

3 Clustering process

The vectors describing the pixel texture are grouped into clusters using a classical K-means algorithm with the help of the Euclidean distance measure.

This classification do not care about the spatial position which is dealt by the connected component extraction process. The separation into two processes allows identifying the different textures independently they belong to the same regions or not.

The number K of clusters is not a prior knowledge. It is iteratively estimated. At the begin of the process, K is fixed to an initial value (6 in experiments). The clustering is done for this value of K and for values $K-1$ and $K+1$. Each clustering is evaluated according to the *trace criterion*: $C_K = trace(S_W^{-1} S_B)$; where S_W is the within-cluster scatter matrix and S_B is the between-cluster scatter matrix ([6], p. 544):

$$\begin{aligned} S_W &= \textstyle\sum_{k=1}^{K} \sum_{j=1}^{J_K} (x_{k,j} - \mu_k)(x_{k,j} - \mu_k)^T \\ S_B &= \textstyle\sum_{k=1}^{K} (\mu_k - \mu)(\mu_k - \mu)^T \end{aligned}$$

with μ_k, the mean of cluster k and μ, the mean of the whole vector set.

If C_K is lower than the trace criteria of the two other clusterings (C_{K-1} and C_{K+1}), the K-partition is the good one and the process stop. Otherwise, if the $(K+1)$-partition (resp. $(K-1)$-partition) minimizes the criterion, the number of cluster is incremented (resp. decremented) by one. The clustering is achieved with this new value and the trace criterion is compared to C_{K+1} (resp. C_{K-1}).

The iteration stops when the criterion minimum is reached. Minimum and maximum numbers of clusters are fixed to bound the process ($K_{min} = 2$ and $K_{max} = 15$ for experiments).

4 Connected component extraction

The clustering process labels each vector according to the class it belongs to. But, no spatial information is taken into account. This allows to identify texture patterns even if they appear in different locations in the image.

To obtain the final image regions, a classical connected component tool is used [3]. Pixels are grouped if they share the same texture label and if they are neighbors.

A 4-connected neighboring is usually used to do this. By propagation, pixels of a region are labeled with the same *region label*.

5 Experiments

Experiments are made on different kinds of images with $J = 3$. The setting of J is reasonnable as more and more pixels share the same caracteristics when the decomposition goes deepest. Experiments confirm this fact. We can see a block effect on some schemas.

All experiment images can be found in color at:

`lita.sciences.univ-metz.fr/~paris/Intensive/`

Figure 3 exhibits the segmentation of a photography of a little rhinoceros. The original image is shown at the top of the figure. Two experiments are conducted from this image.

Firstly, The photography is segmented as explained above. The image is transformed into a pruned Haar wavelet tree. The coefficients of the wave-tree leaves form the texture vectors. These vectors are completed with the energy coefficients (squared values of the pixels) of the chrominance components of the image (components a and b).

The segmentation result is presented in the right column of the figure. The clustering of the vector space delivers two clusters (see top image in right column). And 199 connected components (CC) are extracted from this clustering (see bottom image in right column). Amid all these CCs, The one corresponding to the rhinoceros is relatively homogeneous and easily identifiable.

The left column presents the segmentation for an approximation version of the original image. This approximation is obtained by selecting the approximation delivered by the first level of decomposition. The texture vectors are built with the Haar decomposition of this approximation. In fact, this decomposition correspond to the sub-tree of the wave-tree associated to the approximation. The results are quite similar, but, of course, less precise for the contours of the connected components. Moreover, The segmentation delivers only 33 connected components.

Since the decomposition of the approximation is a sub-tree of the decomposition of the original image, we can use these two segmentations: the clustering and connected component stages use two different vector sets. By this way, the user can select regions from the approximation segmentation or from the original image segmentation.

Experiments in Figures 4 and 5 show similar results. The aborigene and the folk dancers are quite well identified using the approximation segmentations. And the original image segmentations present more precise connected components. While this choice is not of great importance for these images, sometimes it can be. Surprisingly, the duck is difficult to be identified, as shown in Figure 6. Only the segmentation of the approximation offers the possibility to catch all the connected components attached to the duck.

Other experiments are shown in Figure 7. The experiences, presented at lines 4 and 5 in Figure 7, exhibit the weakness of the color characteristic.

6 Conclusion

The first experiments show rough schemata of images. These raw descriptions should help the CBIR user to select the image portions she/he wants to put into her/his request. But, some results fail in describing the objects present in the images. The weakness of the process lies in how to use the multiresolution.

Different kinds of schemata extraction can be investigated:

- One way consists in segmenting each level of resolution. Next, the *segmentation stack* is merged into a schema.

- Another way consists in modifying the euclidean distance, such as :

  1. It is weighted by the level of decomposition; i.e. lower is the resolution level, less important is its impact on the texture definition.
  2. More importance is given to the color components.

One could note that segmented region contours are not precise. However, this is not a major goal in CBIR context since the *silhouettes* of the objects cannot be used to describe them : contours suffer from clutters and occlusions.

Thus, other *invariant* and *robust* features must be defined to characterize regions. For example, a region extension of the *local jets* [13, 16, 26, 27] has to be experimented.

As experiments show, this is a primary study with promising initial results. Several improvements can be experimented to increase its effectiveness. A deepest study can be found at [18].

References

[1] M. S. Allili and D. Ziou. Globally adaptive region information for automatic color-texture image representation. Pattern Recognition Letters, 28(15):1946–1956, 2007.

[2] S. Arivazhagan and L. Ganesan. Texture segmentation using wavelet transform. Pattern Recognition Letters, 24:3197–3203, 2003.

[3] D. H. Ballard and C. M. Brown. Computer Vision. Prentice-Hall, 1982.

[4] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld : Image segmentation using expectation-maximization and application to image querying. IEEE Transaction on Pattern Analysis and Machine Intelligence, 24(8):1026–1038, August 2002.

[5] Y. D. Chun, N. C. Kim, and I. H. Jang. Content-based image retrieval using multiresolution color and texture features. IEEE Tranbsaction on Multimedia, 2008.

[6] R. Duda and P. Hart. Pattern Classification and Scene Analysis - 2nd Ed. Wiley-InterScience, 2000.

[7] J. Gårding and T. Lindeberg. Direct commutation of shape cues using scale-adapted spatial derivative operators. International Journal of Computer Vision, 17(2):163–191, 1996.

[8] J.-M. Geusebroek and A. W. Smeulders. A six-stimulus theory for stochastic texture. International Journal of Computer Vision, 62(1/2):8–16, Appril 2005.

[9] T. Gevers, J. van de Weijer, and H. Stokman. Color Feature Detection, chapter 7. in R. Lukac and K.N. Plataniotis editors, Color Image Processing: Methods and Applications, CRC Press, 2006.

[10] C. Harris and M. Stephens. A combined corner and edge detector. In Proceedings Alvey Conference, pages 189–192, 1988.

[11] K. Jafari-Kouzani and H. Soltanian-Zadeh. Rotation-invariant multiresolution texture analysis using radon and wavelet transform. IEEE Transaction on Pattern Analysis and Machine Intelligence, 14(6):783–795, June 2005.

[12] J. Jiang. Compressed image processing : a new research area and challenge. In Proceedings of Int. Broadcasting Convention, pages 314–319, 1997.

[13] J. Koenderink and A. van Doorn. Image structure. In DAGM-Symposium, pages 3–35, 1997.

[14] S. Liapis, E. Sifakis, and G. Tziritas. Color and texture segmentation using wavelet frame analysis, deterministic relaxation and fast marching algorithms. Journal of Visual Communication and Image Representation, 15(1):1–26, 2004.

[15] T. Lindeberg. Scale-space theory in computer vision. Kluwer academic publisher, 1994.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60:91–110, 2004.

[17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis & Machine Intelligence, 27(10):1615–1630, 2005.

[18] S. Paris. Multiresolution texture segmentation for schematization. Technical Report 2006-102, LITA, 2006.

[19] C.-M. Pun. Rotation-invariant texture feature for image retrieval. *Computer Vision and Image Understanding*, 89:24–43, 2003.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2000(22):888–905, August 8.

[21] M. Unser. Texture classification and segmentation using wavelet frames. IEEE Transaction on Pattern Analysis and Machine Intelligence, 4(11):1549–1560, november 1995.

[22] J. Wang, J. Li, R. Gray, and G. Unsupervised multiresolution segmentation for images with low depth of field. IEEE Transaction on Pattern Recongition and Machine Intelligence, 23(1):85–90, January 2001.

[23] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity : Semantic-Sensitive Integrated Matching For Picture Libraries. IEEE Transaction on Pattern Analysis and Machine Intelligence, 23(9):947–963, September 2001.

[24] S. Wang and J. Siskind. Image segmentation with ratio cut. IEEE Transaction on Pattern Analysis and Machine Intelligence, 25(6):675–690, June 2003.

[25] S. Wang and J. M. Siskind. Image segmentation with minimum mean cut. In Proceedings of the International Conference on Computer Vision, volume 1, pages 517–524, July 2001.

[26] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: An in-depth study. Technical Report RR-5737, INRIA Rhône-Alpes, 665 avenue de l'Europe, 38330 Montbonnot, France, Nov 2005.

[27] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. In Beyond Patches workshop, in conjunction with CVPR, 2006.

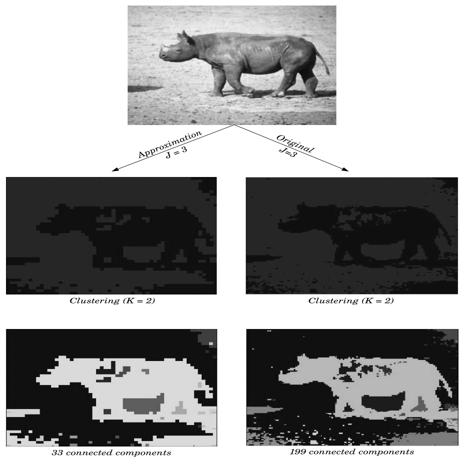

Figure 3. Rhinoceros picture segmentation : at top the original is shown. The left column presents the segmentation of the approximation image, the right one the segmentation of the original image. For each column, the top image shows the texture vector space clustering and the bottom image the connected component result.

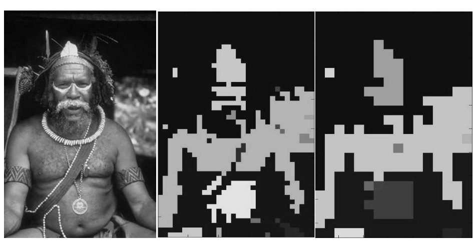

Figure 4. Aborigene picture schema (original, its schema and the schema of its approximation).

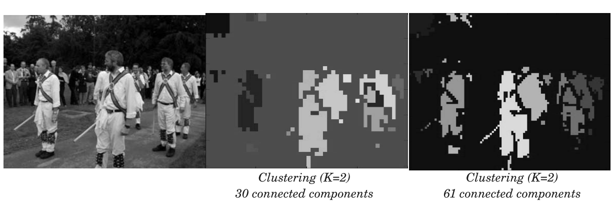

Figure 5. Folk dancer schema.

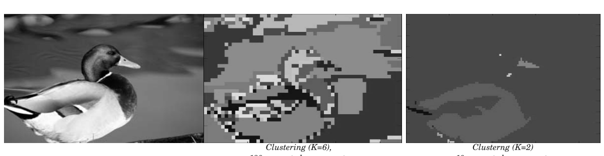

Figure 6. Duck schema.

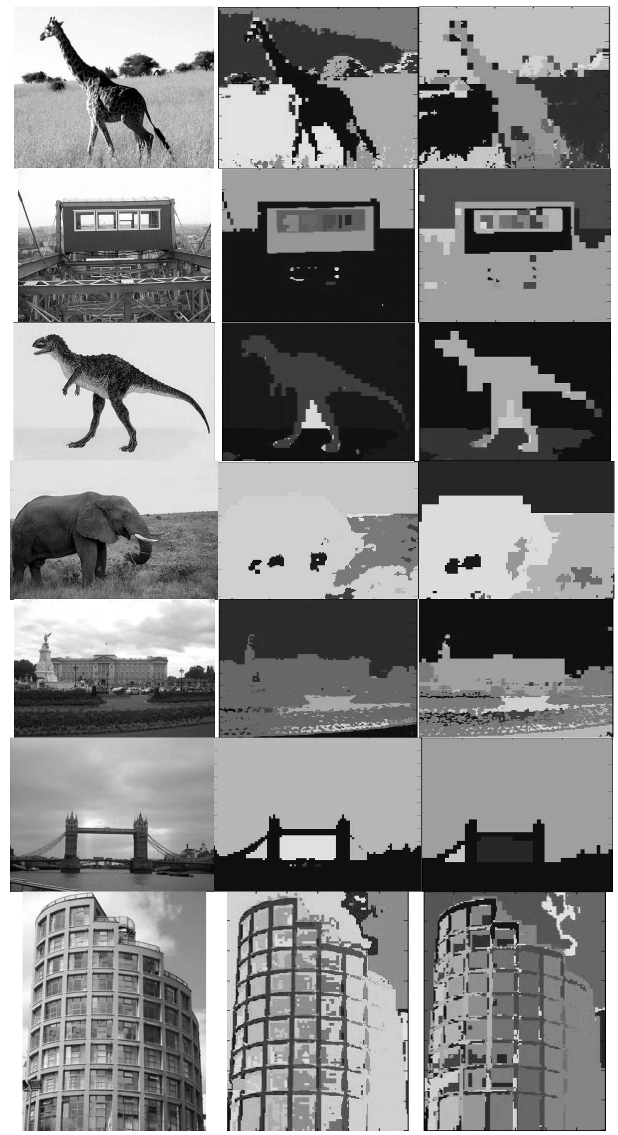

Figure 7. for reach row :At the left the original image, at the middle its schema and at the right the schema of its approximation.